

A method to compute derivatives of functions of large complex matrices

Matthias Pühr* and **Pavel Buividovich**[†]

*Institute of Theoretical Physics, Regensburg University,
93040 Regensburg, Germany*
matthias.puhr@physik.uni-regensburg.de,
pavel.buividovich@physik.uni-regensburg.de

A recently developed numerical method for the calculation of derivatives of functions of general complex matrices, which can also be combined with implicit matrix function approximations such as Krylov-Ritz type algorithms, is presented. An important use case for the method in the context of lattice gauge theory is the overlap Dirac operator at finite quark chemical potential. Derivatives of the lattice Dirac operator are necessary for the computation of conserved lattice currents or the fermionic force in Hybrid Monte-Carlo and Langevin simulations. To calculate the overlap Dirac operator at finite chemical potential the product of the sign function of a non-Hermitian matrix with a vector has to be computed. For non-Hermitian matrices it is not possible to efficiently approximate the sign function with polynomials or rational functions. Implicit approximation algorithms, like Krylov-Ritz methods, that depend on the source vector have to be used instead. Our method can also provide derivatives of such implicit approximations. We show how a generalised deflation prescription can be used to improve the performance of the method, if some eigenvalues and eigenvectors of the matrix being differentiated are known. To show that the method is efficient and well suited for practical calculations we provide test results for the two-sided Lanczos approximation of the finite-density overlap Dirac operator on SU(3) gauge field configurations on lattices with sizes up to 14×14^3 .

*34th annual International Symposium on Lattice Field Theory
24-30 July 2016
University of Southampton, UK*

*Speaker.

[†]This work was supported by the S. Kowalevskaja award from the Alexander von Humboldt foundation.

1. Introduction

Many problems in science and engineering can be formulated with the help of matrix valued functions of matrices [1]. In the field of lattice gauge theories one prominent example for an application of matrix functions is the so-called overlap Dirac operator. The overlap operator respects the Ginsparg–Wilson equation and makes it possible to define chiral symmetry on the lattice. At finite (quark) chemical potential the massless overlap Dirac operator is given by [2]

$$D_{\text{ov}} := \frac{1}{a} (\mathbb{1} + \gamma_5 \text{sgn}[H(\mu)]), \quad (1.1)$$

with the kernel operator $H(\mu) := \gamma_5 D_w(\mu)$, $D_w(\mu)$ is the Wilson–Dirac operator at chemical potential μ , sgn is the matrix sign function and a stands for the lattice spacing. Evaluating the matrix sign function exactly for large matrices is not feasible and for simulations with reasonably large lattices one has to rely on approximation methods to calculate the overlap operator. Efficient approximation methods exist for both Hermitian and non-Hermitian kernel operators. For some applications in lattice gauge theory it is necessary to compute the derivative of the lattice Dirac operator. Examples are the computation of conserved lattice currents or the evaluation of the Fermionic force in HMC and Langevin simulations. For Hermitian kernels one can usually construct a sufficiently good approximation of the overlap operator by using a polynomial or rational function approximation of the sign function. If the kernel is a non-Hermitian matrix on the other hand one generally has to use implicit approximation methods, like Krylov-Ritz type algorithms. For implicit function approximations it is not always clear how to take the derivative of the approximation. Finite difference methods often suffer from severe round-off errors and a naive application of algorithmic differentiation can lead to numerically unstable algorithms [3]. In this contribution we present a numerical differentiation method for functions of general complex matrices and show how a generalised deflation prescription can be used to greatly improve the efficiency of the method. Our method works with any matrix function approximation algorithm and can also be used together with implicit approximation algorithms.

2. Numerical derivatives of matrix functions

Matrix valued functions of a matrix can be defined in several equivalent ways. A very general approach is to define a matrix function via the Jordan decomposition of the matrix. Any complex matrix $A \in \mathbb{C}^{n \times n}$ can be brought to the Jordan normal form

$$X^{-1}AX = J = \text{diag}(J_1, J_2, \dots, J_k), \quad (2.1)$$

where the right hand side is a block diagonal matrix and every block matrix J_i corresponds to an eigenvalue λ_i of A . The J_i are called Jordan blocks and are given by

$$J_i = J_i(\lambda_i) = \begin{pmatrix} \lambda_i & 1 & 0 & \cdots & 0 \\ 0 & \lambda_i & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \lambda_i & 1 \\ 0 & \cdots & 0 & 0 & \lambda_i \end{pmatrix} \in \mathbb{C}^{m_i \times m_i}, \quad (2.2)$$

with $m_1 + m_2 + \dots + m_k = n$. The Jordan matrix J is unique modulo permutations of the blocks, but the transformation matrix X is not. If the Jordan normal form of the matrix is known one can define the matrix function as [1, 4]

$$f(A) := X f(J) X^{-1} = X \operatorname{diag}(f(J_i)) X^{-1}. \quad (2.3)$$

The function of a Jordan block matrix has the form

$$f(J_i) := \begin{pmatrix} f(\lambda_i) & f'(\lambda_i) & \dots & \frac{f^{(m_i-1)}(\lambda_i)}{(m_i-1)!} \\ 0 & f(\lambda_i) & \ddots & \vdots \\ \vdots & \ddots & \ddots & f'(\lambda_i) \\ 0 & \dots & 0 & f(\lambda_i) \end{pmatrix}. \quad (2.4)$$

If A is a diagonalisable matrix the Jordan matrix J becomes a diagonal matrix with the eigenvalues λ_i on the main diagonal and equation (2.3) reduces to the so-called spectral form $f(A) = X \operatorname{diag}(f(\lambda_1), f(\lambda_2), \dots, f(\lambda_n)) X^{-1}$. While the definition (2.4) and the spectral form are very convenient to study matrix functions analytically, they are not well suited for numerical computations. In practice it is however seldom necessary to explicitly know the function of a matrix $f(A)$. In most cases it is sufficient to compute the result $|y\rangle = f(A) |x\rangle$ of the action of the matrix function on a source vector $|x\rangle$.

If the matrix $A = A(t)$ depends on a parameter t then the action of the derivative of the matrix function with respect to t is given by¹ $\partial_t |y\rangle = (\partial_t f(A(t))) |x\rangle$. A theorem by R. Mathias [5] states that the following equation holds under some mild assumptions that are generally fulfilled in lattice QCD calculations:

$$f(\mathcal{B}) = \begin{pmatrix} f(A(0)) & \partial_t f(A(t)) \\ 0 & f(A(0)) \end{pmatrix}, \quad \mathcal{B}(A) := \begin{pmatrix} A(0) & \partial_t A(t) \\ 0 & A(0) \end{pmatrix} \quad (2.5)$$

A notable feature of equation (2.5) is that one does not need to know the explicit form of the function $\partial_t f(A)$ to compute the derivative of $f(A)$. The price to pay is that one has to compute the function f for a block matrix of twice the size of the original matrix A . Since \mathcal{B} is a sparse matrix this is usually not a problem in practical calculations. The big advantage of using equation (2.5) is that the derivative can be computed by applying any approximation algorithm to the left hand side of the following equation:

$$f(\mathcal{B}) \begin{pmatrix} 0 \\ |x\rangle \end{pmatrix} = \begin{pmatrix} \partial_t f(A) |x\rangle \\ f(A) |x\rangle \end{pmatrix} \quad (2.6)$$

An efficient method to compute the matrix sign function for non-Hermitian matrices is the two-sided Lanczos (TSL) method [6–9] and in [3] we demonstrated that the TSL can be used in conjunction with equation (2.6) to compute the derivative of the finite density overlap operator.

3. Deflation

The efficiency of approximation algorithms of matrix functions in general depends on the spectrum of the matrix in question. Evaluating the matrix function is equivalent to computing

¹We assume that the source vector $|x\rangle$ does not depend on t

equation (2.4) for all Jordan blocks and it is obvious that eigenvalues close to a pole or a branch cut of the function f can lead to problems in a numerical algorithm. The sign function has a discontinuity along the imaginary axis and it has been shown that the approximations of the matrix sign function can be greatly improved by a so called “deflation” [6–8]. In the deflation procedure the eigenvectors corresponding to eigenvalues close to zero are projected out from the source vector $|x\rangle$ and the sign function is evaluated exactly on the subspace spanned by these eigenvectors:

$$\text{sgn}(A)|x\rangle = \underbrace{\sum_{i=1}^m \text{sgn}(\lambda_i) |R_i\rangle \langle L_i|}_{\text{exact}} |x\rangle + \underbrace{\text{sgn}(A) P_m^n}_{\text{approximation}} |x\rangle, \quad (3.1)$$

where $|R_i\rangle$ and $\langle L_i|$ are the right and left eigenvectors of A corresponding to the eigenvalue λ_i and $P_m^n = \sum_{i=m+1}^n |R_i\rangle \langle L_i|$ is a projector. The standard deflation procedure (3.1) relies on the spectral decomposition of A and works only for diagonalisable matrices. The block matrix \mathcal{B} is in general not diagonalisable and in the following we briefly describe how to generalise the deflation procedure to non-diagonalisable matrices. For an in-depth derivation of all results in this section we refer to [10]. The Jordan normal form of \mathcal{B} is given by

$$\mathcal{X}^{-1} \mathcal{B} \mathcal{X} = \mathcal{J} \quad \text{where} \quad \mathcal{J} = \begin{pmatrix} J_1 & 0 & 0 & \dots & 0 \\ 0 & J_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 0 & J_n \end{pmatrix} \quad \text{with} \quad J_i := \begin{pmatrix} \lambda_i & 1 \\ 0 & \lambda_i \end{pmatrix}. \quad (3.2)$$

The transformation matrices \mathcal{X} and \mathcal{X}^{-1} can be constructed analytically in terms of the eigenvalues and (left and right) eigenvectors of the matrix A and their derivatives:

$$\mathcal{X}^{-1} = \begin{pmatrix} (\langle L_1|, \langle \partial_t L_1|) \\ \partial_t \lambda_1(0, \langle L_1|) \\ \vdots \\ (\langle L_n|, \langle \partial_t L_n|) \\ \partial_t \lambda_n(0, \langle L_n|) \end{pmatrix} \quad \text{and} \quad \mathcal{X} = \left(\begin{pmatrix} |R_1\rangle \\ 0 \end{pmatrix}, \frac{1}{\partial_t \lambda_1} \begin{pmatrix} |\partial_t R_1\rangle \\ |R_1\rangle \end{pmatrix}, \dots, \begin{pmatrix} |R_n\rangle \\ 0 \end{pmatrix}, \frac{1}{\partial_t \lambda_n} \begin{pmatrix} |\partial_t R_n\rangle \\ |R_n\rangle \end{pmatrix} \right) \quad (3.3)$$

We can now use the known analytic expressions for the Jordan decomposition of \mathcal{B} to devise a generalised deflation procedure. Instead of projecting out the eigenvectors corresponding to the eigenvalues λ_i close to zero, we project out the subspace corresponding to the Jordan block J_i . For the sake of a compact notation let $|X_i\rangle$ denote the columns of \mathcal{X} and $\langle \tilde{X}_i|$ the rows of \mathcal{X}^{-1} . Then $\sum_{i=1}^{2n} |X_i\rangle \langle \tilde{X}_i| = \mathbb{1}$ and we can define the projectors $\mathcal{P}_m = \sum_{i=1}^m |X_i\rangle \langle \tilde{X}_i|$ and $\tilde{\mathcal{P}}_m := \mathbb{1} - \sum_{i=1}^m |X_i\rangle \langle \tilde{X}_i|$, such that every vector $|\psi\rangle \in \mathbb{C}^{2n \times 1}$ can be written as $|\psi\rangle = \mathcal{P}_m |\psi\rangle + \tilde{\mathcal{P}}_m |\psi\rangle$. Likewise we can write $f(\mathcal{B})|\psi\rangle = \mathcal{X} f(\mathcal{J}) \mathcal{X}^{-1} \mathcal{P}_{2l} |\psi\rangle + f(\mathcal{B}) \tilde{\mathcal{P}}_{2l} |\psi\rangle$, where we use the projectors for the l eigenvalues $\lambda_1, \dots, \lambda_l$ closest to zero and equation (2.3). The function of the Jordan blocks is given by equation (2.4) and by exploiting the bi-orthogonality of $|X_i\rangle$ and $\langle \tilde{X}_i|$ we finally

get

$$f(\mathcal{B})|\psi\rangle = \underbrace{f(\mathcal{B})\tilde{\mathcal{P}}_{2l}|\psi\rangle}_{\text{approximation}} + \underbrace{\sum_{i=1}^l [f(\lambda_i)(|X_{2i-1}\rangle\langle\bar{X}_{2i-1}| \psi\rangle + |X_{2i}\rangle\langle\bar{X}_{2i}| \psi\rangle) + (\partial_i f(\lambda_i))|X_{2i-1}\rangle\langle\bar{X}_{2i}| \psi\rangle]}_{\text{exact}}. \quad (3.4)$$

This looks almost like the deflation procedure for diagonalisable matrices, with the difference that we have to use the vectors that make up the Jordan transformation matrices instead of the eigenvectors. Moreover because of the non-diagonal structure of the Jordan blocks there are additional “mixing terms” proportional to $\partial_i f(\lambda_i)$. The sign function is piece-wise constant and the derivative terms vanish, so that the generalised deflation prescription for the sign function is given by $\text{sgn}(\mathcal{B})|\psi\rangle = \text{sgn}(\mathcal{B})\tilde{\mathcal{P}}_{2l}|\psi\rangle + \sum_{i=1}^{2l} \text{sgn}(\lambda_i)|X_i\rangle\langle\bar{X}_i|\psi\rangle$. Note that it is sufficient to know the first l columns (rows) of \mathcal{X} (\mathcal{X}^{-1}) to perform the deflation. The required l eigenvalues and eigenvectors of A can be computed with the Arnoldi method and in our calculations we use the ARPACK package for this task. The calculation of the derivative of the eigenvectors is a bit more involved. The derivative of a right eigenvector is given by $|\partial_i R_j\rangle = \sum_{i \neq j} \frac{|R_i\rangle\langle L_i|(\partial_i A)|R_j\rangle}{\lambda_j - \lambda_i}$ and a similar equation holds for the derivative of the left eigenvector. The summation over all eigenvectors can be avoided by using the fact that $\sum_{i \neq j}^n \frac{|R_j\rangle\langle L_j|}{\lambda_i - \lambda_j} = \sum_{i=1}^{(j-1)} \frac{|R_j\rangle\langle L_j|}{\lambda_i - \lambda_j} + (\lambda_i - A)^{-1} P_l^n$.

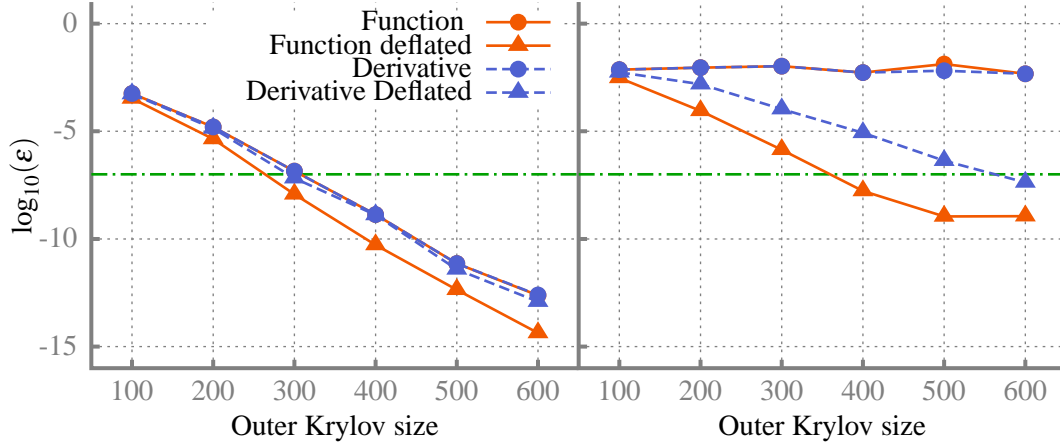


Figure 1: Results for the error of the nested TSL approximation (inner Krylov subspace size 100) of the sign function and its derivative with and without deflation. The 40 eigenvalues closest to zero are used to deflate the sign function. For the deflation of the derivative we use 6 (2) eigenvalues for $V = 14 \times 14^3$ ($V = 6 \times 18^3$). On the left plot $V = 6 \times 18^3$ and $\mu = 0.230$ and on the right $V = 14 \times 14^3$ and $\mu = 0.300$. The dash-dotted green line marks an error of 10^{-7} .

4. Numerical Results

To test the efficiency our numerical differentiation method and the generalised deflation prescription we compute derivatives of the overlap Dirac operator at finite chemical potential. Our quenched $SU(3)$ gauge configurations are generated with the tadpole-improved Lüscher–Weisz action [11]. We use two different parameter sets, a lattice volume $V = 14 \times 14^3$ with $\beta = 8.1$ corresponding to a temperature of $T = 113$ MeV and $V = 6 \times 18^3$ with $\beta = 8.45$ corresponding to

$T = 346$ MeV. The critical temperature for the deconfinement phase transition for the Lüscher-Weisz action is around 300 MeV and with our choice of parameters we have configurations for both phases. The derivatives are taken with respect to an external $U(1)$ lattice gauge field $\Theta_V(x)$. To approximate the matrix sign function we use a nested version of the TSL [9] with one level of nesting. The error of the approximation is estimated using the fact that the square of the sign function is the identity and we define the (relative) error of the sign function evaluation as $\epsilon_A = \frac{\|\text{sgn}(A)^2|\psi\rangle - |\psi\rangle\|}{2\|\psi\rangle\|}$, where the factor two in the denominator is used because we have to apply the TSL approximation twice to compute the square of the sign function. The error of the derivative is computed in the same way, substituting \mathcal{B} for A and the sparse vector $(0, |\phi\rangle)^T$ for $|\psi\rangle$.

In Figure 1 we show the error of the function and the derivative as a function of the (outer) Krylov subspace size. We find numerically that in the high temperature phase the spectrum of the kernel operator already has a large gap around zero and consequently the deflation does not lead to a big improvement. In the low temperature phase on the other hand the gap in the spectrum is considerably smaller and we can clearly see that the deflation greatly improves the efficiency of the TSL algorithm. Not only is the error for the deflated approximation much smaller at a given Krylov subspace size, we also observe that the improvement of the error is much bigger as the Krylov subspace size is increased.

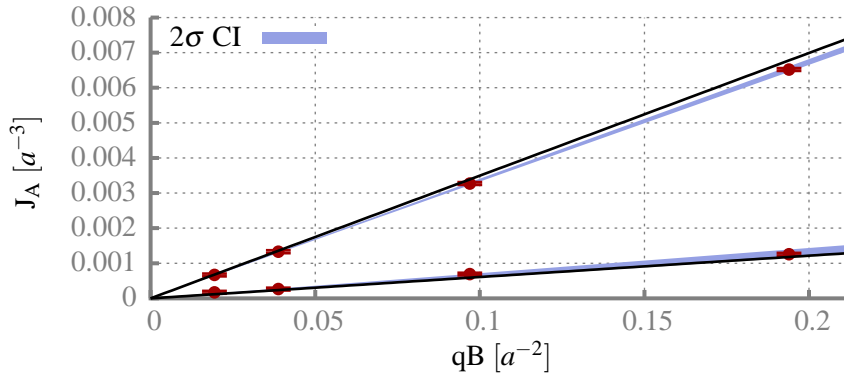


Figure 2: Results for the axial current in the chirally symmetric phase of QCD. The lattice size is 6×18^3 and $\beta = 8.45$. The chemical potential is $\mu = 0.040$ and $\mu = 0.230$ (steeper slope) in lattice units. Red dots mark the results for J^A and the blue bands the 2σ confidence intervals for σ_{CSE} . The black lines denote the result for non-interacting fermions in the continuum.

5. Conclusion and Outlook

In this work we have presented a method to take numerical derivatives of functions of general complex matrices. We have shown how a generalised deflation prescription can be used to improve the efficiency of the method and we have tested our method on quenched $SU(3)$ gauge configurations. Our main motivation to develop this method was the need to compute conserved lattice currents for the overlap Dirac operator in order to study anomalous transport effects in dense QCD. In particular we are interested in the Chiral Separation Effect (CSE) [12], where an external magnetic field B induces an axial current parallel to the field in a plasma of chiral fermions with finite chemical potential μ : $J_i^A = \sigma_{\text{CSE}} B_i$. The proportionality constant σ_{CSE} is called the chiral separation conductivity and for free fermions in the continuum it is given by $\sigma_{\text{CSE}} = qN_c N_f \mu / 2\pi^2$. It

is expected that this relation also holds in the chirally symmetric phase of QCD, but there could be corrections if chiral symmetry is spontaneously broken [13]. On the lattice the conserved axial current reads as $\langle J^A(x) \rangle = \text{tr} \left(D_{\text{ov}}^{-1} \frac{\partial D_{\text{ov}}}{\partial \theta_{x,\mu}} \gamma_5 \right)$. To prove that our method is well suited for practical calculations in Figure 2 we show first results for the axial current and σ_{CSE} in the chirally symmetric phase of QCD for two different values of the chemical potential. The conserved axial current is not renormalised in lattice calculations and we can directly compare our data with continuum results. Our error bars are very small and we find a remarkably good agreement between our results and the continuum value of σ_{CSE} . A more detailed analysis of the CSE in both phases of QCD is work in progress and the results will be published elsewhere [14].

References

- [1] Nicholas J. Higham. *Functions of Matrices: Theory and Computation*, SIAM, 2008.
- [2] J. Bloch, T. Wettig, *Overlap Dirac Operator at Nonzero Chemical Potential and Random Matrix Theory*, Phys. Rev. Lett. **97** (2006), 012003 [hep-lat/0604020].
- [3] M. Pühr and P. V. Buividovich. *A method to calculate conserved currents and fermionic force for the Lanczos approximation to the overlap Dirac operator*, PoS(LATTICE2014)047.
- [4] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press Baltimore and London, 3rd edition, 1996.
- [5] R. Mathias, *A Chain Rule for Matrix Functions and Applications*, SIAM J. Matrix Anal. & Appl. **17** (1996), 610–620.
- [6] J. Bloch et al. *Short-recurrence Krylov subspace methods for the overlap Dirac operator at nonzero chemical potential*, Comput.Phys.Commun. **181** (2010), 1378–1387, [0910.1048].
- [7] J. Bloch, T. Breu, T. Wettig, *Comparing iterative methods to compute the overlap Dirac operator at nonzero chemical potential*, PoS(LATTICE 2008)027.
- [8] J. Bloch, T. Wettig, A. Frommer, and B. Lang. *An iterative method to compute the overlap Dirac operator at nonzero chemical potential*, PoS(LATTICE 2007)169.
- [9] J. Bloch, S. Heybrock, *A nested Krylov subspace method to compute the sign function of large complex matrices*, Comput. Phys. Commun. **182** (2011), 878–889 [0912.4457].
- [10] M. Pühr and P. V. Buividovich. *A numerical method to compute derivatives of functions of large complex matrices and its application to the overlap dirac operator at finite chemical potential*, Comput. Phys. Commun. **208** (2016), 135–148, [1604.08057].
- [11] M. Lüscher, P. Weisz, *On-shell improved lattice gauge theories*, Comm. Math. Phys. **97** (1985), 59–77.
- [12] M. A. Metlitski and A. R. Zhitnitsky. *Anomalous axion interactions and topological currents in dense matter*, Phys.Rev.D **72** (2005), 045011, [hep-ph/0505072].
- [13] G. M. Newman and D. T. Son. *Response of strongly-interacting matter to magnetic field: some exact results*. Phys.Rev.D **73** (2006), 045006, [hep-ph/0510049].
- [14] M. Pühr and P. V. Buividovich. *A first principles study of the Chiral Separation Effect in quenched finite-density QCD*, unpublished.